REMARKS

The specification has been amended to correct minor errors. These corrections are of a clerical nature and do not add "new matter".

Claims 6, 10, 11, and 13 have been amended to further particularly point out and distinctly claim subject matter regarded as the invention. The text of claim 12 is unchanged, but its meaning is changed because it depends from amended claim 10.

Claim 6 has been amended to improve antecedent basis.

New claims 14-26 also particularly point out and distinctly claim subject matter regarded as the invention.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current Amendment. The attached page is captioned "Version with Markings to Show Changes Made."

The 35 U.S.C. § 102 Rejection

According to M.P.E.P. § 2131, "[a] claim is anticipated [under 35 U.S.C. §102(a), (b), and (e)] only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." It goes on to state that "[t]he elements must be arranged as required by the claim..."

Claims 1, 5, and 9 stand rejected under 35 U.S.C. § 102(b) as being allegedly anticipated by *Orita* (US 5,163,147). This rejection is respectfully traversed.

Generally, the Office Action states that *Orita* discloses all of the claim elements. However, each and every element as set forth in the present claims are not found in *Orita*. Furthermore, the various combinations of elements proposed by the Office Action are never arranged by *Orita* in the same manner as proposed by the Office Action or as required by the present claims. Specifically, several of the citations are duplicated, are only one line of disclosure, or are to the Summary of the Invention section. These citations do not fully explain or enable the disclosure of *Orita* or explain the basis of the rejection. Although it is possible for the Office Action to rely on what is "well known" as a basis for rejection, to be proper this must

be taken as Official Notice and the modification of the cited reference must be fully explained to comply with § 2131. The modifications can not be assumed or implied, but must be explicit.

For discussion purposes, a review of *Orita* will first be presented. *Orita* was originally filed in Japanese and the combination of the English translation and the approach of the disclosure leaves something to be desired in terms of clarity. According to FIG. 1 of *Orita*, the system hardware includes a work station 10, a host computer 11, and a storage unit 12 connected by a network. The host computer 11 includes a CPU 13 and a RAM 14. Prior to operation of the system, operator profile (OP) information 12c, environmental profile (EP) information 12d, user programs 12e, and user files 12f are loaded onto the storage unit 12. "Access protection information 12a (not shown) is included in each of the user programs 12e and each of the user files 12f." (col. 3, lines 7-9) (See col. 2, line 53 through col. 3, line 9, among others.) The operation of the system is explained with respect to the steps shown in FIGS. 2 and 3. In steps S1 through S5, the user logon operation is described. (Col. 3, lines 10-32) In step S6, the host computer 11 reads the OP information 12c and stores it into area 14a of RAM 14 and reads EP information 12d and stores it into area 14b, as appropriate. This is a simple data relocation exercise to make the data more convenient for the host computer to access. Not all of the data is moved and none of the data is altered. (See col. 3, line 33 through col. 4, line 22.) The operation then performs the first of a two part permission verification process. In steps S7-S9, the operation verifies whether the user is permitted to execute a particular user (job) program 12e and if so, the host computer 11 reads the user program 12e and stores it into storage area 14c. This again is a simple relocation exercise to make the program more convenient for the host computer to access. Not all of the programs are moved and the program is not altered. (See col. 3, lines 23-45.) In steps S10 and S11, the user program is executed and it requests access to one or more of the user files 12f. The operation then performs the second of the two part permission verification process. In steps S12-S15, "[t]he host computer 11 *compares* the content...of the access type which is to be executed by the user program...with the contents of access protection information 12a of [the] corresponding [user] files [to which access is requested], and permits

the [user program to] access [the user file] if [the access type of the user program and the access protection information of the user file] coincide with each other..." (emphasis added) (Col. 4, lines 60-67) (See col. 4, line 46 through col. 5, line12.) In steps S16 and S17, "[w]hen the user program is completed, [the operation determines] whether or not the system is continuously used." (Col. 5, lines 13 and 14) (See col. 5, lines 13-21.) This essentially completes the operation of the system. *Orita* concludes that "[t]herefore, the security can be attained not for each user as in the conventional case but for each operation based on [(1)] the user program and [(2)] the file access thereof." (Col. 5, lines 45-47)

The two part permission verification process is central to the disclosure of *Orita*. The Office Action overlooks the two part process by selectively focusing only on the second part. The second part depends on the first part and is not distinct from it. To ignore the first part is to change the principle of operation of *Orita* which is not allowed. Further, *Orita* only discloses "comparing" the access type. The access protection information 12a is set by some other process prior to operation of the system and is never changed. It is not sufficient to argue that *Orita* could change the information if he never discloses that he actually does. The static nature of the access protection information is contrary to the present claims 1, 5, and 9 where, as variously claimed, the active file security status is "changeable" from a first type to a second type and the active file security status is actually "changed" from the first type to the second type.

Given the above, *Orita* can not be said to anticipate the present claims.

Claims 10-13 stand rejected under 35 U.S.C. § 102(e) as being allegedly anticipated by *Scott et al.* (US 5,987,123). This rejection is respectfully traversed.

Generally, the Office Action states that *Scott* discloses all of the claim elements. However, similar to above, *Scott* discloses a multi level system and the Office Action selectively focuses on only one of the levels. *Scott* states that "...the described embodiment of the present invention incorporates two or more levels or signatures. A file access must satisfy both levels before it is allowed by the file system." (Col. 6, lines 56-59) To ignore one level is to change the principle of operation of *Scott*. Further, *Scott* is directed to providing "...a truly automatic

and transparent method of checking and authenticating software and data in a computer system." (Col. 1, lines 30-31) That is to say that *Scott* wants the user to be "secure" in the knowledge that the software or data are trustworthy. This is a different sense of the term secure. Further still, *Scott* does not disclose all of the claim elements as presently amended.

Given the above, *Scott* can not be said to anticipate the present claims.

The 35 U.S.C. § 103 Rejection

Claims 2 and 6 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Orita* (US 5,163,147) in view of *Subramaniam et al.* (US 5,519,507). Claims 3, 4, 7, and 8 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over *Orita* (US 5,163,147) and *Subramaniam et al.* (US 5,519,507) in view of *Testin et al.* (US 4,776,038). These rejections are respectfully traversed.

Generally, the Office Action states that *Orita* discloses or suggests most of the claim elements and that *Subramaniam* and *Testin* disclose or suggest the rest of the claim elements. However, claims 2-4 depend from claim 1 and claims 6-8 depend from claim 5 and the arguments presented above with respect to claims 1 and 5 apply equally here. The addition of *Subramaniam* and *Testin* does noting to refute those arguments. Thus the cited references can not be said to render the present claims obvious.

In view of the above, it is respectfully asserted that the claims are now in condition for allowance.

Request for Allowance

In view of the foregoing, reconsideration and an early allowance of this application are earnestly solicited.

If any matters remain which could be resolved in a telephone interview between the Examiner and the undersigned, the Examiner is invited to call the undersigned to expedite resolution of any such matters.

Respectfully submitted,
THELEN, REID, & PRIEST LLP

Dated: November 21, 2002

David B. Ritchie
Reg. No. 31,562

Thelen, Reid, & Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
(408) 292-5800

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

The paragraph beginning at page 10, line 20 has been amended as follows:

-- In accordance with the present invention, deleting a secure file may not be an atomic operation because it would be undesirable to allow clients the ability to delete a secure file unless the client was authorized to do so. Delete operations must be able to differentiate between files that should be able to be deleted (non-secure files and secure files being deleted by an authorized client) and those that [shouldn't] should not be able to be deleted (secure files attempted to be deleted by an unauthorized client). Therefore, the atomic delete operation must be designed to check the fixed file security status of a file before deleting it. Only files with an associated fixed file security status of type "operations allowed" will be deleted atomically. A new apparatus and method has been developed for the deletion of a secure file. --

The paragraph beginning at page 12, line 4 has been amended as follows:

-- The set delete-on-close request may also be used to delete non-secure files, though the atomic delete operation is available and easier. However, there is no need for validation in this case. --

IN THE CLAIMS

Claims 6, 10, 11, and 13 have been amended as follows:

6. (Amended Once) The method of claim 5, wherein the apparatus has a third memory associated with the file, said third memory storing a delete-on-close status, said third memory initially storing a first value and changeable to a second value wherein said first value indicates

that the file will not be deleted upon closing and said second value indicates that the file will be deleted upon closing, the method further comprising:

receiving a delete-on-close request from said client;

changing said delete-on-close status from said first value to said second value; and

deleting the file upon closing.

10. (Amended Twice) A method for creating a secure file on a file system of a router, said router further including a request handler and a verification routine, [the] said method comprising:

receiving from a user at said request handler an open for write call for a file that does not exist at the time [the] said call is received;

recognizing at said request handler that [the] said file does not exist at the time [the] said call is received;

creating with said request handler a file entry for said file;

receiving from said user at said request handler an authorization credential;

authenticating with said verification routine the privileges of [the] said user;

recognizing with said request handler a [the] combination of [a] said user sending an open for write call for a file that does not exist at the time [the] said call is received and [an] said authorization credential that is authenticated; and

creating in said file system said [a] secure file having a fixed file security status being of a first type.

11. (Amended Once) The method of claim 10, further comprising:

setting a memory location associated with [the] said file and in said file system of said router to a value indicating that [the] said file is a secure file.

13. (Amended Twice) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for creating a secure file on a file system, the program of instructions including a request handler routine and a verification routine, the method comprising:

receiving from a user with said request handler routine an open for write call for a file that does not exist at the time [the] said call is received;

recognizing with said request handler routine that [the] said file does not exist at the time [the] said call is received;

creating with said request handler routine a file entry for said file;

receiving from said user with said request handler routine an authorization credential;

authenticating with said verification routine the privileges of [the] said user;

recognizing with said request handler routine a [the] combination of [a] said user sending an open for write call for a file that does not exist at the time [the] said call is received and [an] said authorization credential that is authenticated; and

creating [a] said secure file having a fixed file security status being of a first type.